OIPE
FEB 0 2 2004
JC24
PATENT & TRADEMARK OFFICE

#19
S. Zand
2/10/04

RECEIVED
FEB 0 5 2004
Technology Center 2100

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

| | | | | |
|---|---|---|---|---|
| Title | : | PROTOCOL TECHNOLOGY FOR CLIENT/SERVER ENVIRONMENT | | |
| Inventors | : | David T. Sulcer; Donna S. Lowe-Cleveland; Lawrence M. Ackner | | |
| Serial No | : | 09/336,424 | | |
| Filed | : | 6/17/1999 | Examiner Hai V. Nguyen | |
| Docket | : | 149-0112US | Group Art Unit 2142 | |

Commissioner for Patents
Mail Stop Appeal Briefs-Patent
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF TRANSMISSION (37 C.F.R. 1.8)

This paper is being transmitted to the Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 via U.S. First Class mail on January 28, 2004.

Joyce Cranford
Joyce Cranford

## APPEAL BRIEF

This is an appeal from the rejection of claims 1-52 in the Advisory Office Action dated 12/15/2003 and Final Office Action dated 10/14/2003. For the convenience of the Board of Appeals and Interferences, a table of contents for the remainder of this Appeal Brief follows on the next page.

02/04/2004 EFLORES 00000129 501922 09336424
01 FC:1402 330.00 DA

RECEIVED
FEB 0 5 2004
Technology Center 2100

## TABLE OF CONTENTS

## REAL PARTY IN INTEREST

The real party in interest in the above referenced patent application is BMC Software, Inc. of Houston, Texas.

## RELATED APPEALS AND INTERFERENCES

To the present knowledge of Appellants' representative, there are currently no related appeal or interference proceedings that will directly affect, or be directly affected by, or have a bearing on, the Board's decision in the present Appeal.

## RELATED APPLICATIONS

To the best of Appellants' knowledge, there are no known related applications.

## STATUS OF THE CLAIMS

In the aforementioned Office Actions, claims 1-52 were examined. Claims 1-52 stand rejected under 35 U.S.C. § 102(e) and 103(a), as being unpatentable in view of U.S. Patent 5,973,696 to *Agranat* et al. (hereinafter *Agranat*).

## STATUS OF AMENDMENTS FILED SUBSEQUENT TO FINAL REJECTION

A Reply was filed on 11/20/2003 responsive to the Final Office Action mailed 10/14/2003. An interview with the Examiner was held on 11/19/2003. An Advisory Action was mailed on 12/15/2003. No amendments have been filed subsequent to the Final Office Action.

## SUMMARY OF THE PRESENT INVENTION

The methods and systems disclosed in the current application are applicable to software in client/server environments, especially to application software that is a combination of client software and server software wherein the client software executes on a client computer and the inter-dependent server software runs on a server computer. (Page 10, Line 7 – 8.) According to the current invention, application software can be configured dynamically based on the client/server environment such that the client-server software applications can be implemented and/or execute more efficiently. (Page 5-6)

In the current application, the interactivity between client software and server software is controlled by three fundamental characteristics: definitions, state changes and commands. (Page 10, Lines 19 – 22.)

The first of the three fundamental characteristic is "definition." A definition is any fixed aspect of the environment associated with the application. For example, a data entry form in an application may have fixed categories. The fixed categories are "definitions." (Page 14, Lines 17 – 19.) In another example, an application may be displayed in a Graphic User Interface (GUI) having a fixed color scheme. The fixed color scheme is "definition." (Page 15, lines 11-16.)

The second of the three fundamental characteristic is "state change." State is the matter associated with the programmable environment that is not fixed; that is – the application allows for its modification during execution. (Page 18, Lines 9 – 12.) For example, the data entry in a category may be inserted/deleted/changed. Those data represent possible "state change."

The third of the three fundamental characteristic is "command." Commands are directives from the local software (running on either on a client or a server) that directly or indirectly cause the remote software (running on the server or the client) to undertake some action or function. (Page 22, Line 7 – 9.) For example, a "Get Form" directive is a "command." A "Get Form" sent from the client software to the server software causes the server software to generate a Form and send the Form to the client software for further use. (Page 24.)

In one embodiment of current invention as claimed, a single message contains all three characteristics: definition, state change, and command. (Page 30, Lines 14 – 20.) The three fundamental characteristics are transferred between the client and the server, and are processed in a particular order: definition first, state change second, and command third. (Page 14, Lines 3 – 5.)

In another embodiment, synchronization is maintained between the client software and the server software. That is, a sending device (i.e., client software or server software) will not send additional messages until it receives some form of acknowledgment by the receiving device that it has received the prior message. (Page 31, Line 21 – Page 32, Line 2). The client software running on the client and the server software running on the server are inter-dependent.

## ISSUES

There are six independent claims: 1, 12, 26, 37, 51 and 52. The following comprise the issues at hand:

1) Whether the Examiner has identified all claim elements of claim 1 in *Agranat*.
2) Whether the Examiner has identified all claim elements of claim 12 in *Agranat*.

| **Rejection** | **References Cited** | **Claim Numbers Affected** |
|---|---|---|
| 35 U.S.C. § 102 (e) | *Agranat* | 1-50 |
| 35 U.S.C. § 103 (a) | *Agranat* | 51-52 |

## GROUPING OF CLAIMS

Claims 1-11, 26-36 and 51 stand or fall together.

Claims 12-25, 37-50 and 52 stand or fall together.

## ARGUMENTS

### 1. Summary of the disclosure of *Agranat*

*Agranat* is directed to an "Embedded Web Server." The disclosure relevant to the present application is essentially the generic operation of a web browser, a web server and their interaction.

The communication between a browser and a server, as shown in Figures 11 or 12 of *Agranat*, is generally in an interactive fashion, i.e., each transmission contains only one request or instruction. A web page, such as shown in Figure 9 of *Agranat*, may be sent from a server to a browser and displayed in the browser. The web page may include text or graphics to be displayed in the browser and instructions for a user to supply information or command. But nowhere does *Agranat* teach, discuss or fairly suggest a sequence by which a web page is processed. As would be known by those of ordinary skill in the art, prior art web browsers do not care which sequence to display a web page. Also, the displayed web page is generally unrelated to either the browser or the web server.

A web browser application on a client computer and a web server application on a server are completely independent, i.e., the operation or existence of a web browser application on a client computer is completely irrelevant to the operation or existence of a web server application on a server computer. The browser may choose to interact with a first web server and can interact with a second web server regardless whether a response is received from the first server.

### 2. Requirement for rejections under 35 U.S.C. § 102 and 35 U.S.C. § 103

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros.* v. *Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). Further, the "identical invention must be shown in as complete detail as is contained in the patent claim" (*Richardson* v. *Suzuki Motor Co.*, 868 F.2d 1226, 1236 (Fed. Cir.), *cert. denied*, 493 U.S. 853 (1989)) and the "elements must be arranged as in the claim under review" (*In re Bond*, 910 F.2d 831, 832 Fed. Cir. 1990), reh'g denied, 1990 U.S. App. LEXIS 19971 (Fed. Cir. 1990)). *See* M.P.E.P. 2131, 8th Ed., Rev. 1, Feb. 2003, page 2100-70.

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *See* M.P.E.P. 2143, 8th Ed., Rev. 1, Feb. 2003, page 2100-124.

3. Whether the Examiner has identified each and every element of claim 1 in *Agranat*

The Examiner asserts that *Agranat* discloses each and every element of claim 1 such that *Agranat* anticipates claim 1. The Appellant disagrees. Claim 1 reads:

1. A method, comprising:
   a) [1] *receiving* a [3] *message* that was [2] *sent from a remote device*, [4] *said message comprising* a [5] *definition*, a [6] *state change*, and a [7] *command*, said definition defining [10] a *fixed aspect* of [9] *a user environment*, said state change describing a change in [11] *a non-fixed aspect* of said user environment, said command being a [12] *directive* that causes a function to be performed; and,
   b) [8] *processing* said definition *before* said state change and said command are processed,
      processing said state change before said command is processed, and
      *processing* said command.

(Numbers with "[ ]" are added for easy identification in later discussion.)

The Appellant submits that the Examiner has not identified *each and every* of the above claim elements or their equivalents in the office actions (and as discussed below, the Examiner *cannot* identify all of the above claim elements). It appears that the Examiner has correlated various objects disclosed in *Agranat* to various objects required in claim 1, but ignored most of the inter-relationships between the various objects as required by the claims. The inter-relationships are also part of the claimed elements.

| Claim element | Objects/elements disclosed in *Agranat* | Appellant' Comments |
|---|---|---|
| [1]*receiving* | Fig. 12, Browser | Assume, for the sake of argument, that *Agranat's* Browser corresponds to the device performing the act of "*receiving.*" |
| [2]*sent from a remote device* | Fig. 12, EmWeb/server | Assume, for the sake of argument, that *Agranat's* EmWeb/server corresponds to the recited "*remote device*," and that the EmWeb/server "*sends*" a "*message.*" |
| [3]*message* | Fig. 12, HTTP response message 1207, 1215 | There are two separate responses 1207 and 1215 shown in Figure 12. They cannot be the same message. Assume, for the sake of argument, one of the HTTP responses 1207 or 1215 is the "*message.*" |
| [4]*said message comprising* | | "*Said message*" must contain at least the three recited characteristics: [5]*definition*, [6]*state change* and [7]*command.* |
| [5]*Definition* | Fig. 12, form name 1201, 1203 | Form name, as shown in Figure 12 may be part of HTTP requests 1201 or 1203, but Form name is not part of either HTTP response 1207 or 1215, which is the "*message.*" Therefore, Form name cannot be the "*definition*" in the "*message.*"<br>Further, as shown in Figure 12, HTTP request 1201 is from the Browser to the EmWeb/Server, and HTTP |

| | | |
|---|---|---|
| | | request 1203 is from the EmWeb/Server to the Application. Neither 1201 or 1203 is "*received*" by the Browser or "*sent from*" the EmWeb/Server. So neither can be the "*message*." For this reason, Form name cannot be the "*definition*." |
| [6]*State change* | Fig. 12, default value 1205 | Default value 1205 is not part of HTTP response 1207 or 1215 ("*message*") as shown in Figure 12. Therefore, Default value 1205 cannot be the recited "*state change*." Further, Default value 1205 is sent by the "Application" to the "EmWeb/Server", not between the EmWeb/Server ([2]) and the Browser([1]) as asserted by the Examiner earlier and required by claim 1. |
| [7]*Command* | Fig. 12, Submit 1211 | Submit 1211 is not part of HTTP response 1207 or 1215 ("*message*") as shown in Figure 12. Therefore, Submit 1211 cannot be the recited "*command*."<br>Further, Submit 1211 is sent by the EmWeb/Server ([2]) to the Application, not between the EmWeb/Server ([2]) and the Browser ([1]) as asserted by the Examiner earlier and required by claim 1. |
| [8] *processing Definition/ state change/ command* | Fig. 12, 1203, 1205, then 1209-1215 | The items asserted by the examiner: 1203, 1205, 1207, 1209-1215 are transmitted/received at different times, by three different objects: Browser, EmWeb/Server and Application. They are not part of either 1207 or 1215 (the "*message*"). Therefore, they cannot be any of the [5]*definition*, [6]*state change*, or [7]*command*. Their processing is irrelevant to the claim. |

From the above table, it is clear that the Examiner failed to identify, in *Agranat*, at least "[4] *said message comprising* a [5] *definition*, a [6] *state change*, and a [7] *command*," and "[8] *processing* . . .," i.e., the elements 4, 5, 6, 7 and 8 required by claim 1. Because of this failure, *Agranat* cannot anticipate claim 1. The Appellant respectfully requests the Board vacate the Examiner's rejection and allow claims 1-11, 26-36 and 51 to issue.

In addition, claim 1 is directed to a Client/Server environment where applications executing on client computer and server computer interact with each other, regarding the [9] *user environment*. In contrast, *Agranat* is directed to a web server and an improved graphical user interface. In a web page written in HTML, there can be many different objects, such as static text, graphics, hyperlinks, drop-down list, buttons or dynamic text, etc. Each of these objects are related to the content of the web page, rather than the web browser, the web server or other aspects of the [9] *user environment*. There is no teaching in *Agranat* that any of those objects in a web page represent two aspects of a [9] *user environment* -- one [10] *fixed* ([5]*definition)* and one [11] *non-fixed* ([6] *state change*). Assuming, for the sake of argument, *Agranat* teaches objects in a web page corresponding to [10] *fixed* and [11] *non-fixed* aspects of [9] the *user environment*, *Agranat* fails to teach that these two objects, together with another object [12] *directive* ([7] *command*) are present in a single webpage ([3] *message*). There is no teaching that such arrangement, i.e., a web page including one [10] *fixed* and one [11]*non-fixed* aspects of a [9] *user environment*, and a [12] *directive*, is even desirable. Further, there is absolutely no disclosure in *Agranat* that these objects are processed in a particular sequence. To the contrary, for example, the various objects in a web page shown in Figure 9 in *Agranat* do not appear to be arranged in any particular sequence. There is no teaching or fair suggestion that it is desirable to process any objects on a web page in a particular sequence. There is no motivation to do so in the context of a web page.

*Agranat* does not disclose the claim elements of [10] *fixed* and [11]*non-fixed* aspects of a [9] *user environment*, and a [12] *directive*, within a [3] *message*. Therefore, *Agranat* does not anticipate claim 1. Furthermore, *Agranat* does not meet the three criteria to establish a *prima facie* case of obviousness. Therefore, *Agranat* does not render claim 1 obvious.

The Appellant respectfully requests the Board to allow claims 1-11, 26-36 and 51 to issue.

4. Whether the Examiner has identified all claim elements of claim 12 in *Agranat*

The Examiner has rejected claim 12 in view of *Agranat*. The Appellant respectfully disagrees. Claim 12 reads:

> 12. A method executed by a local device that cooperatively operates with a remote device in order to implement an application software program, said device separated from said remote device by a network, said method comprising:
>
> a) recognizing that a dependence on said remote device exists, said dependence being a need for said remote device to perform some act, said act allowing said application software program to continue to operate; and
>
> b) sending a message to said remote device, said message comprising either a definition, a state change, a command or some combination thereof, said definition defining a fixed aspect of a said application software program, said state change describing a change in a non-fixed aspect of said application software program, said command being a directive that causes said remote device to perform said act.

Claim 12 is directed to an embodiment that is slightly different from the embodiment recited in claim 1. In claim 12, the emphasis is on the "*dependence*" between the two devices on which an application software program is executing. The *dependence* is "*a need for said remote device to perform some act, said act allowing said application software program to continue to operate*."

The Examiner asserts that "the presence an EMWEB_STRING tag 1107" in Figure 11 in *Agranat* corresponds to the "*dependence*" required in claim 12. The assertion is beside the point. *Agranat's* EMWEB_STRING tag in a document located on the *server* may cause the *server* to perform some operation and then return some *dynamic* data to the browser. (*Agranat*, Col. 16, lines 28-44). It does not mean that there is a *need* for the server to perform the requested operation in order for the *browser* to "*continue operate*." Even if the server does not perform the act specified by the EMWEB_STRING tag, the browser can still continue to operate. For example, visit or interact with other servers.

It is well known to those of ordinary skills in the relevant art that, by design, the operability of Internet browsers (or servers) does not depend upon the operation or availability of any particular server (or browser). In prior art browser/server environments, there is no *dependence*. Operations of the browser and the server are mutually independent: a browser may operate, for example visit a second website, whether the first web server is operating or not. Similarly, the first server can service requests from another browser whether the first browser is operating or not. The teaching of *Agranat*, which is directed to web server, is contrary to the teaching of claim 12 regarding *dependence*.

From the above discussion, it is clear that *Agranat* does not disclose at least the *dependence* element of claim 12. Further, *Agranat* does not meet the three criteria to establish a *prima facie* case of obviousness regarding claim 12 (see discussion above regarding claim 1). Therefore, *Agranat* does not anticipate claim 12, nor does *Agranat* render claim 12 obvious. Accordingly, the Appellant respectfully requests the Board vacate the Examiner's rejections and allow claims 12-25, 37-50 and 52 to issue.

## SUMMARY AND CONCLUSIONS

In light of the aforementioned differences, Appellant submits that the claimed invention is patentably distinct over the cited prior art, and respectfully requests the Board reverse the Examiner's rejections as to claims 1-52 and allow the claims to issue.

1/28/2004
Date

Liangang (Mark) Ye
Reg. No. 48,276
Email: mye@counselIP.com
Wong, Cabello, Lutsch, Rutherford & Brucculeri, L.L.P.
20333 SH 249, Suite 600
Houston, Texas 77070
Voice: 832-446-2407
Facsimile: 832-446-2424

Appendix: Claims on Appeal

1. (Previously presented) A method, comprising:

a) receiving a message that was sent from a remote device, said message comprising a definition, a state change, and a command, said definition defining a fixed aspect of a user environment, said state change describing a change in a non-fixed aspect of said user environment, said command being a directive that causes a function to be performed; and,

b) processing said definition before said state change and said command are processed,

processing said state change before said command is processed, and

processing said command.

2. (Previously presented) The method of claim 1 wherein said command further comprises an express command.

3. (Original) The method of claim 2 wherein said express command corresponds to clicking a mouse button.

4. (Original) The method of claim 2 wherein said express command corresponds to hitting an enter key.

5. (Original) The method of claim 2 wherein said express command corresponds to selecting an option from a menu.

6. (Previously presented) The method of claim 1 wherein said command is an instruction command.

7. (Original) The method of claim 6 wherein said instruction command corresponds to getting a form.

8. (Previously presented) The method of claim 1 wherein said state change corresponds to a new value in a form.

9. (Previously presented) The method of claim 1 wherein said definition corresponds to the definition of a form.

10. (Previously presented) The method of claim 1 wherein said processing said definition further comprises updating a definition record associated with a GUI understanding with said definition.

11. (Previously presented) The method of claim 1 wherein said processing said state change further comprises updating a state record associated with a GUI understanding with said state change.

12. (Previously presented) A method executed by a local device that cooperatively operates with a remote device in order to implement an application software program, said device separated from said remote device by a network, said method comprising:

a) recognizing that a dependence on said remote device exists, said dependence being a need for said remote device to perform some act, said act allowing said application software program to continue to operate; and

b) sending a message to said remote device, said message comprising either a definition, a state change, a command or some combination thereof, said definition defining a fixed aspect of a said application software program, said state change describing a change in a non-fixed aspect of said application software program, said command being a directive that causes said remote device to perform said act.

13. (Original) The method of claim 12 wherein said command (if any) further comprises an express command.

14. (Original) The method of claim 13 wherein said express command corresponds to clicking a mouse button.

15. (Original) The method of claim 13 wherein said express command corresponds to hitting an enter key.

16. (Original) The method of claim 13 wherein said express command corresponds to selecting an option from a menu.

17. (Original) The method of claim 12 wherein said command (if any) is an instruction command.

18. (Original) The method of claim 17 wherein said instruction command corresponds to getting a form.

19. (Original) The method of claim 12 wherein said state change (if any) corresponds to a new value in a form.

20. (Original) The method of claim 12 wherein said definition (if any) corresponds to the definition of a form.

21. (Original) The method of claim 12 further comprising updating a GUI understanding by processing said definition (if any) before processing said state change (if any).

22. (Original) The method of claim 21 wherein said processing said definition (if any) further comprises updating a definition record associated with said GUI understanding with said definition (if any).

23. (Original) The method of claim 21 wherein said processing said state change (if any) further comprises updating a state record associated with said GUI understanding with said state change (if any).

24. (Previously presented) The method of claim 12 further comprising:

a) receiving an acknowledgement message from said remote device, said acknowledgement message comprising either a second definition, a second state change, a second command or some combination thereof; and

b) processing said second definition (if any) before said second state change (if any) and before said second command (if any); and

c) processing said second state change (if any) before processing said second command (if any).

25. (Original) The method of claim 24 further comprising translating said definition (if any), said state change (if any) and said command (if any) to a GUI.

26. (Previously presented) A device having a processor and a memory coupled to the processor, the memory having stored therein sequences of instructions which, when executed by the processor, cause the processor to perform the acts of:

a) receiving a message that was sent from a remote device, said message comprising a definition, a state change, and a command, said definition defining a fixed aspect of a user environment, said state change describing a change in a non-fixed aspect of said user environment, said command being a directive that causes a function to be performed; and

b) processing said definition before said state change and said command are processed,

processing said state change before said command is processed, and

processing said command.

27. (Previously presented) The device of claim 26 wherein said command further comprises an express command.

28. (Original) The device of claim 27 wherein said express command corresponds to clicking a mouse button.

29. (Original) The device of claim 27 wherein said express command corresponds to hitting an enter key.

30. (Original) The device of claim 27 wherein said express command corresponds to selecting an option from a menu.

31. (Previously presented) The device of claim 26 wherein said command is an instruction command.

32. (Original) The device of claim 31 wherein said instruction command corresponds to getting a form.

33. (Previously presented) The device of claim 26 wherein said state change corresponds to a new value in a form.

34. (Previously presented) The device of claim 26 wherein said definition corresponds to the definition of a form.

35. (Previously presented) The device of claim 26 wherein said processing said definition further comprises updating a definition record associated with a GUI understanding with said definition.

36. (Previously presented) The device of claim 26 wherein said processing said state change further comprises updating a state record associated with a GUI understanding with said state change.

37. (Previously presented) A local device that cooperatively operates with a remote device in order to implement an application software program, said device separated from said remote device by a network, said device having a processor and a memory coupled to the processor, the memory having stored therein sequences of instructions which, when executed by the processor, cause the processor to perform the acts of:

a) recognizing that a dependence on said remote device exists, said dependence being a need for said remote device to perform some act, said act allowing said application software program to continue to operate; and

b) sending a message to said remote device, said message comprising either a definition, a state change, a command or some combination thereof, said definition defining a fixed aspect of a said application software program, said state change describing a change in a non-fixed aspect of said application software program, said command being a directive that causes said remote device to perform said act.

38. (Previously presented) The local device of claim 37 wherein said command (if any) further comprises an express command.

39. (Previously presented) The local device of claim 38 wherein said express command corresponds to clicking a mouse button.

40. (Previously presented) The local device of claim 38 wherein said express command corresponds to hitting an enter key

41. (Previously presented) The local device of claim 38 wherein said express command corresponds to selecting an option from a menu.

42. (Previously presented) The local device of claim 37 wherein said command (if any) is an instruction command.

43. (Previously presented) The local device of claim 42 wherein said instruction command corresponds to getting a form.

44. (Previously presented) The local device of claim 37 wherein said state change (if any) corresponds to a new value in a form.

45. (Previously presented) The local device of claim 37 wherein said definition (if any) corresponds to the definition of a form.

46. (Previously presented) The local device of claim 37 further comprising updating a GUI understanding by processing said definition (if any) before processing said state change (if any).

47. (Previously presented) The local device of claim 46 wherein said processing said definition (if any) further comprises updating a definition record associated with said GUI understanding with said definition (if any).

48. (Previously presented) The local device of claim 37 wherein said processing said state change (if any) further comprises updating a state record associated with said GUI understanding with said state change (if any).

49. (Previously presented) The local device of claim 37 wherein said sequence of instructions further cause the processor to implement the steps of:

a) receiving an acknowledgement message from said remote device, said acknowledgement message comprising either a second definition, a second state change, a second command or some combination thereof; and

b) processing said second definition (if any) before said second state change (if any) and before said second command (if any); and

c) processing said second state change (if any) before processing said second command (if any).

50. (Previously presented) The local device of claim 49 further comprising translating said definition (if any), said state change (if any) and said command (if any) to a GUI.

51. (Previously presented) A machine readable medium having stored thereon sequences of instructions, which when executed by a processor, cause the processor to perform a method, said method comprising:

a) receiving a message that was sent from a remote device, said message comprising a definition, a state change, and a command, said definition defining a fixed aspect of a user environment, said state change describing a change in a non-fixed aspect of said user environment, said command being a directive that causes a function to be performed; and

b) processing said definition before said state change and said command are processed,

processing said state change before said command is processed, and

processing said command.

52. (Previously presented) A machine readable medium to be executed by a local device that cooperatively operates with a remote device in order to implement an application software program, said device separated from said remote device by a network, said machine readable medium having stored thereon sequences of instructions, which when executed by a processor, cause the processor to perform the acts of:

a) recognizing that a dependence on said remote device exists, said dependence being a need for said remote device to perform some act, said act allowing said application software program to continue to operate; and

b) sending a message to said remote device, said message comprising either a definition, a state change, a command or some combination thereof, said definition defining a fixed aspect of a said application software program, said state change describing a change in a non-fixed aspect of said application software program, said command being a directive that causes said remote device to perform said act.